MovieLens Project

Mahaman Sani SAHIROU ADAMOU

2022-06-13

Contents

1 Introduction

The purpose of this project is to create a **movie recommendation system** using the MovieLens dataset. This involve building a model that predicts the rating of a movie from known informations (**predictors** or **independent variables**). We will use a light ten(10) millions ratings version of the MovieLens dataset to make the computation a little easier. This light dataset is produced by a piece of code provided by **EDX**. We will first look at the structure of the data, visualize it and then progressively build a model that will reach a targeted accuracy.

Diffrent models will be **trained** on a nine(9) millions ratings set and evaluated on a one(1) million ratings set, the **validation** set, based on the **RMSE (Root Mean Squared Error)**. Finally, the best model will be chosen.

2 Analysis

2.1 Brief overview of the dataset

The code chunc provided by EDX generate a 9 millions rows training set and 1 million rows validation set, both sets of 6 variables.

- **userId** `<integer>` : unique identification number for the user,
- **movieId** `<numeric>` : unique identification number for the movie,
- **rating** `<numeric>` : 5-stars rating grade given by a user to a movie,
- **timestamp** `<integer>` : date and time the rating was given to a movie by a user,
- **title** `<character>` : movie title (not unique), also contains the year the movie was released,
- **genres** `<character>` : genres associated with the movie, pipe-separated values.

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

«rating» is the **dependent variable** that will be estimated by our models. The **5** remaning variables are **independent variables** or **predictors**.

There is no missing values (NAs) in both traning and validation sets.

No zero 0-star rating was given in the edx training set.

2.2 Further data preparation

We see through the table attached to the paragraph above that:

- in addition to the title of the movie, the **title** variable also contains the movie year of realease that can be useful for the model. These two pieces of information should be separated,
- **genres** are a pipe-separated list of values that should also be separated,
- we can extract more readable an useful informations (**year** and **month**) from the **timestamp** variable.

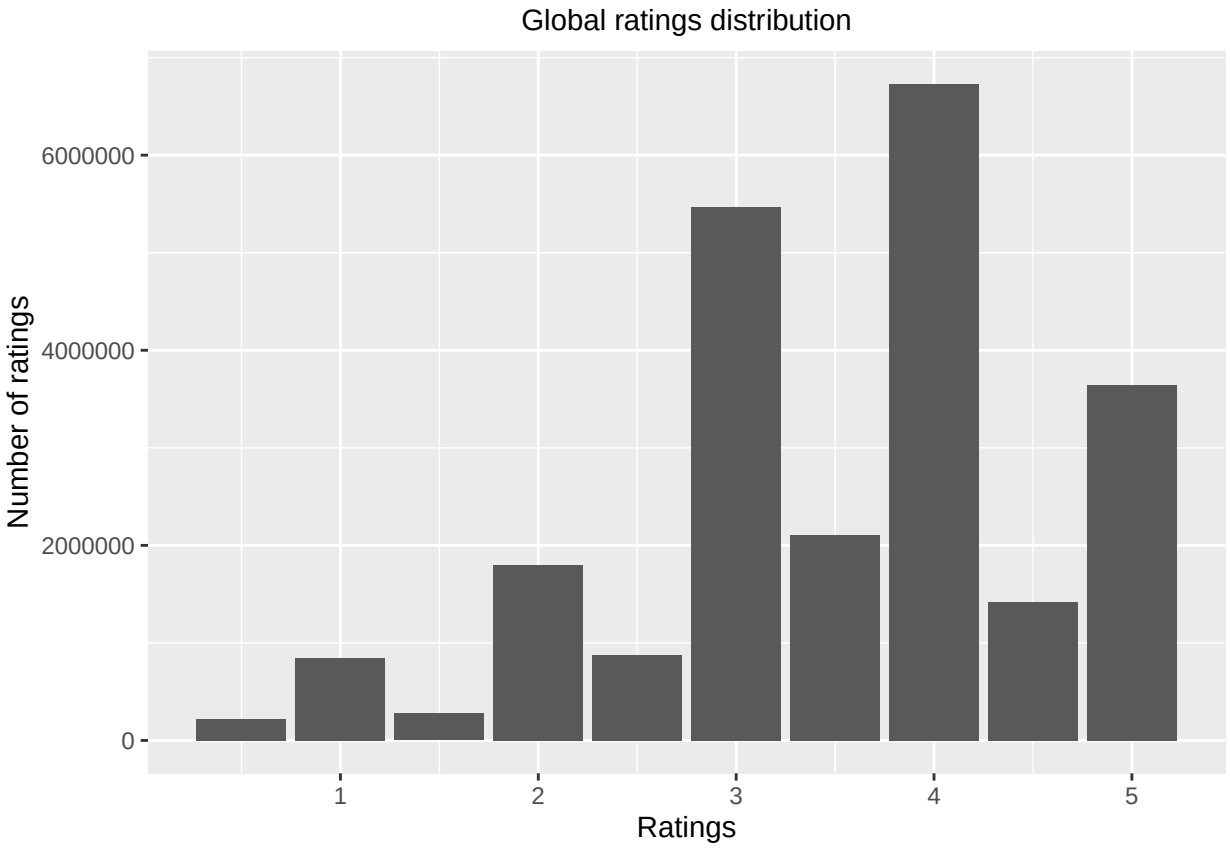
After these transformations, we end up with *longer* and litle *wider training* and *validation* sets:

movieId	movieTitle	movieReleaseYear	genre	userId	rating	ratingMonth	ratingYear
122	Boomerang	1992	Comedy	1	5	8	1996
122	Boomerang	1992	Romance	1	5	8	1996
185	Net, The	1995	Action	1	5	8	1996
185	Net, The	1995	Crime	1	5	8	1996
185	Net, The	1995	Thriller	1	5	8	1996
292	Outbreak	1995	Action	1	5	8	1996

We can see that in the training set, **69878** distinct users rated **10677** distinct movies, that theoretically corresponds to more than **746** millions possible combinations (versus 10 millions rows in the training set), so, not every user has rated every movie.

2.3 Exploratory data analysis

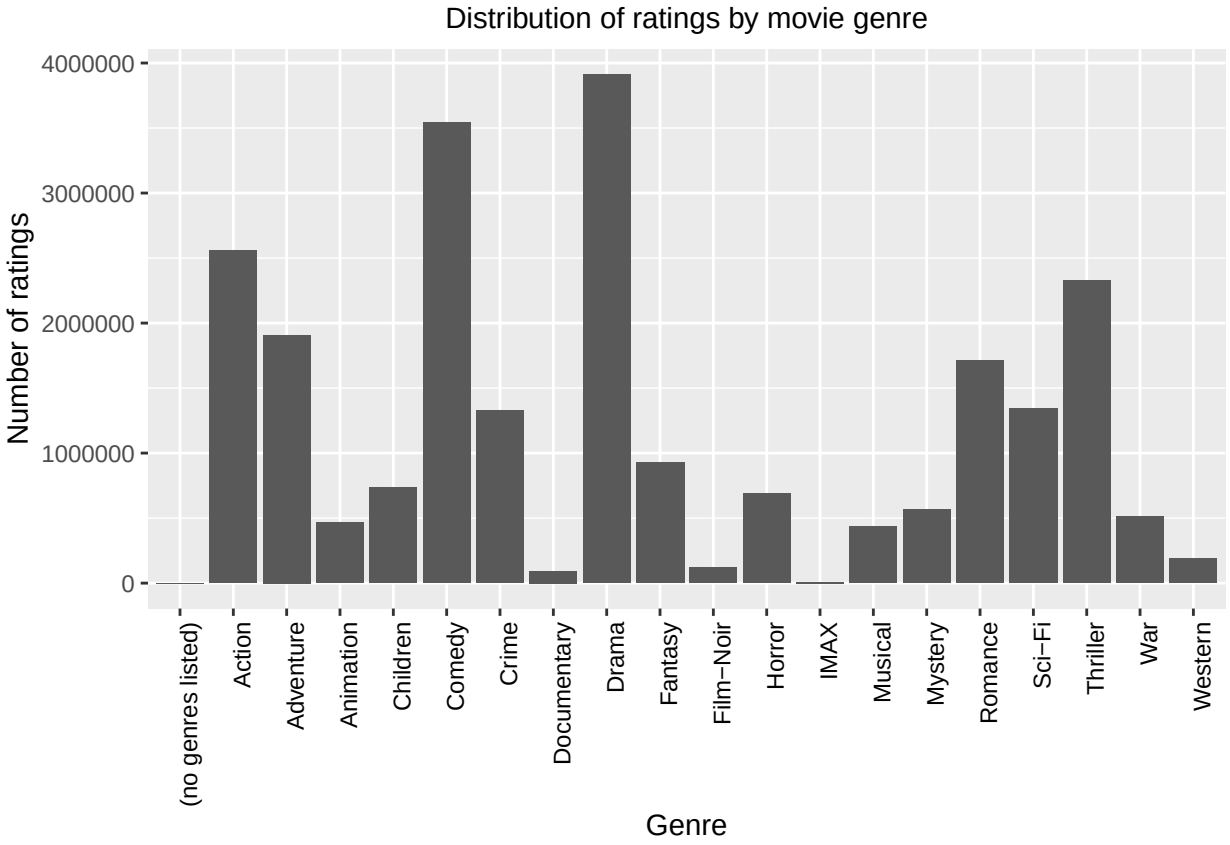
2.3.1 Overview



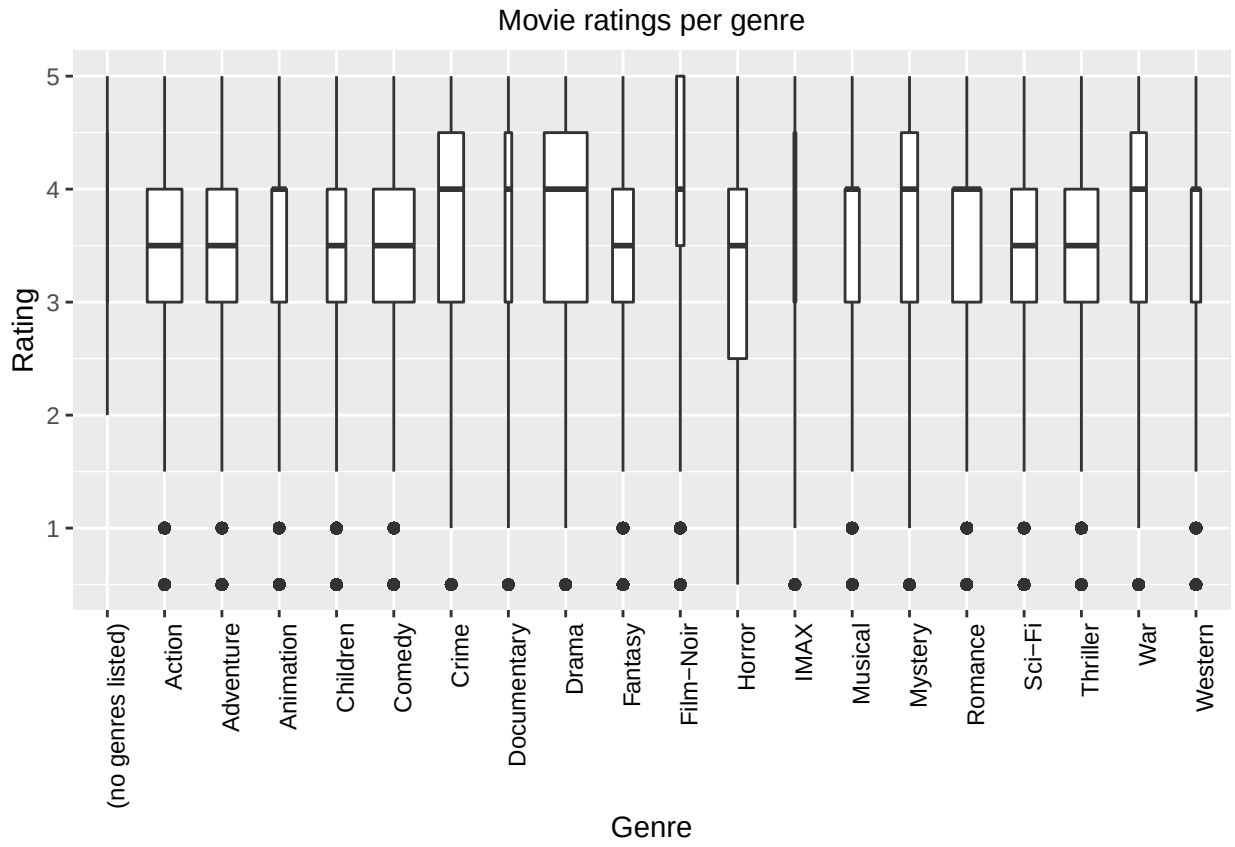
half-star ratings (0.5, 1.5, 2.5, 3.5 and 4.5) are less common (**20.95%** of ratings in the training set) than **full-star** ratings (1, 2, 3, 4 and 5).

The ratings distribution is left-skewed : we have more positive ratings: **59.47%** of ratings in the training set are above 2.5 on a 5 stars rating scale. This is probably due to the fact that users are more likely to give a rating when they already wached and liked the movie.

2.3.2 Ratings distribution per Genre

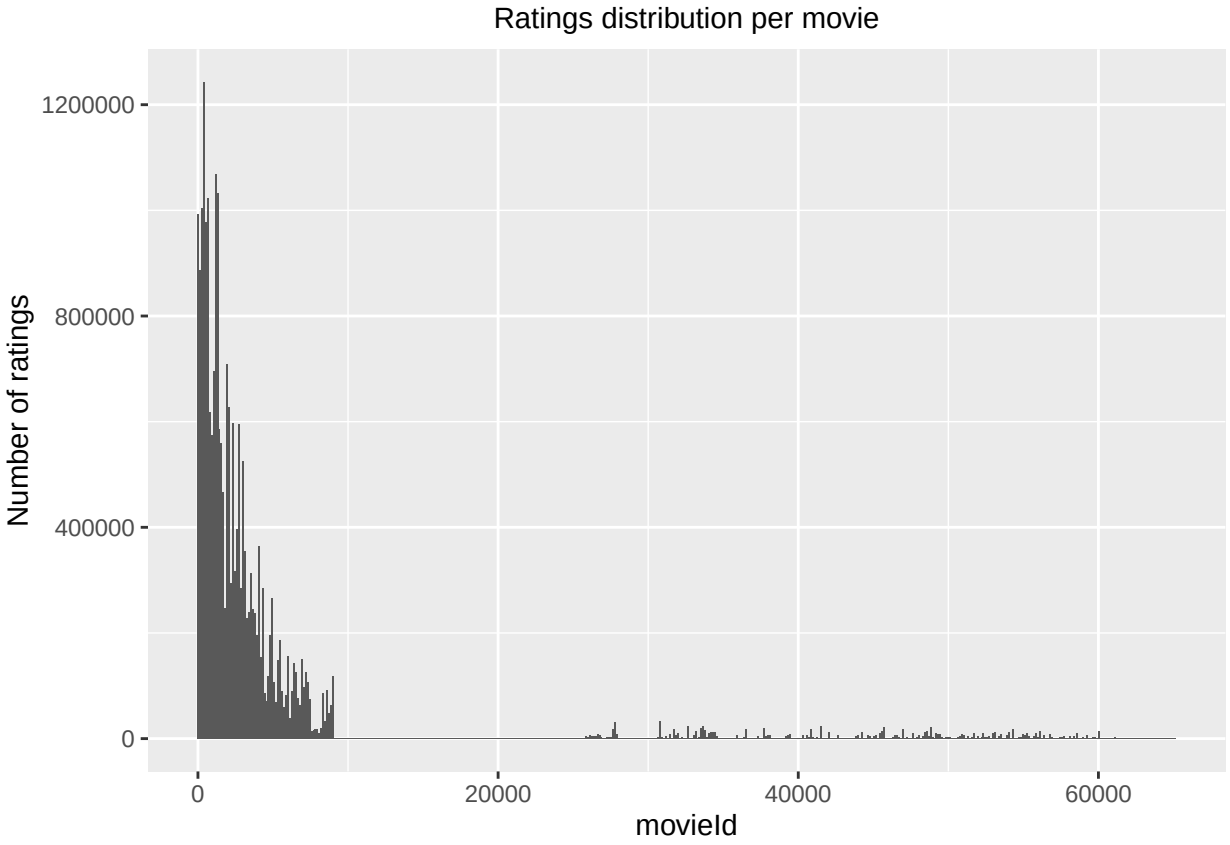


The **genre** variable contains **20** distinct values of genres movies are characterized in. The most popular rated genre types are **Drama** and **Comedy**.

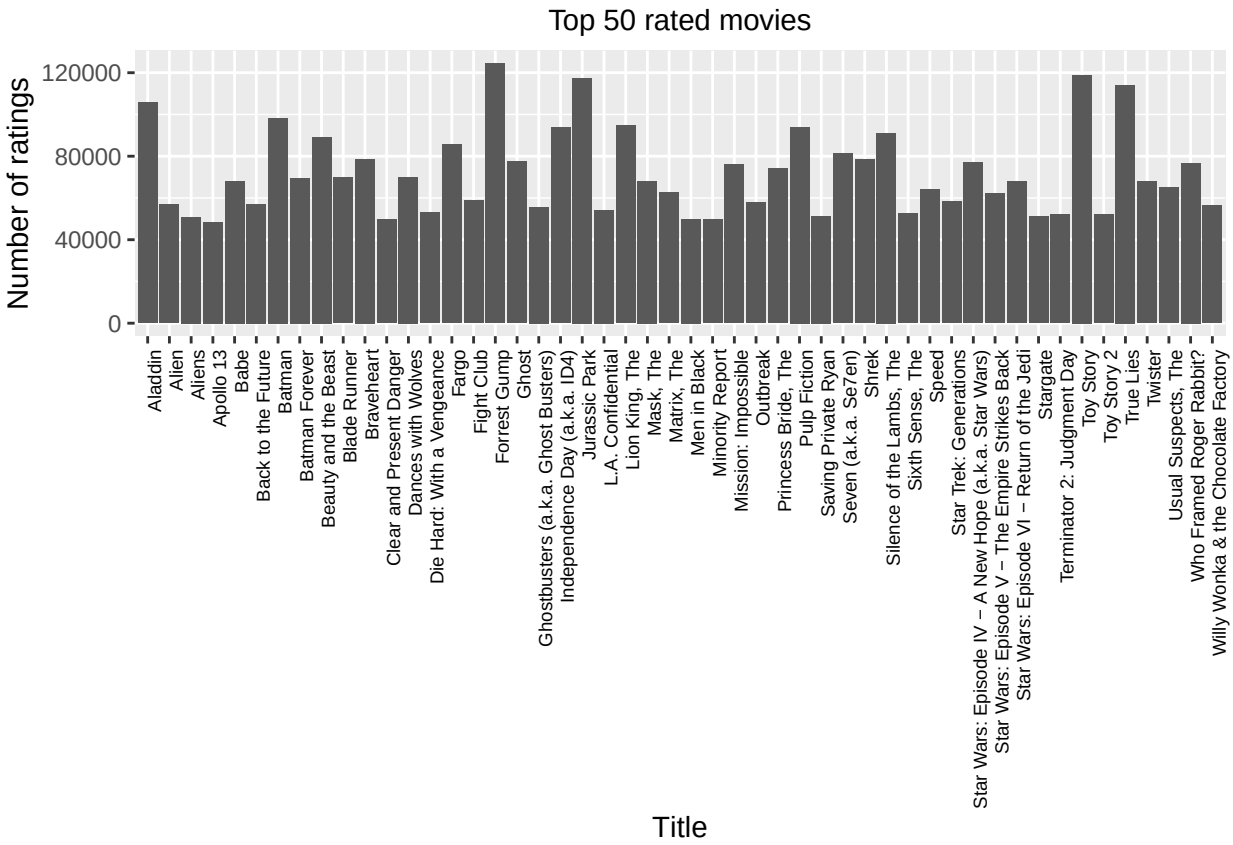


From the graph above, we can see that the ratings appear to be different between genres. **Film-Noir** and **Documentary** are some of the better rated genre types, while **Sci-Fi** and **Horror** are some of the worst rated.

2.3.3 Ratings distribution per Movie

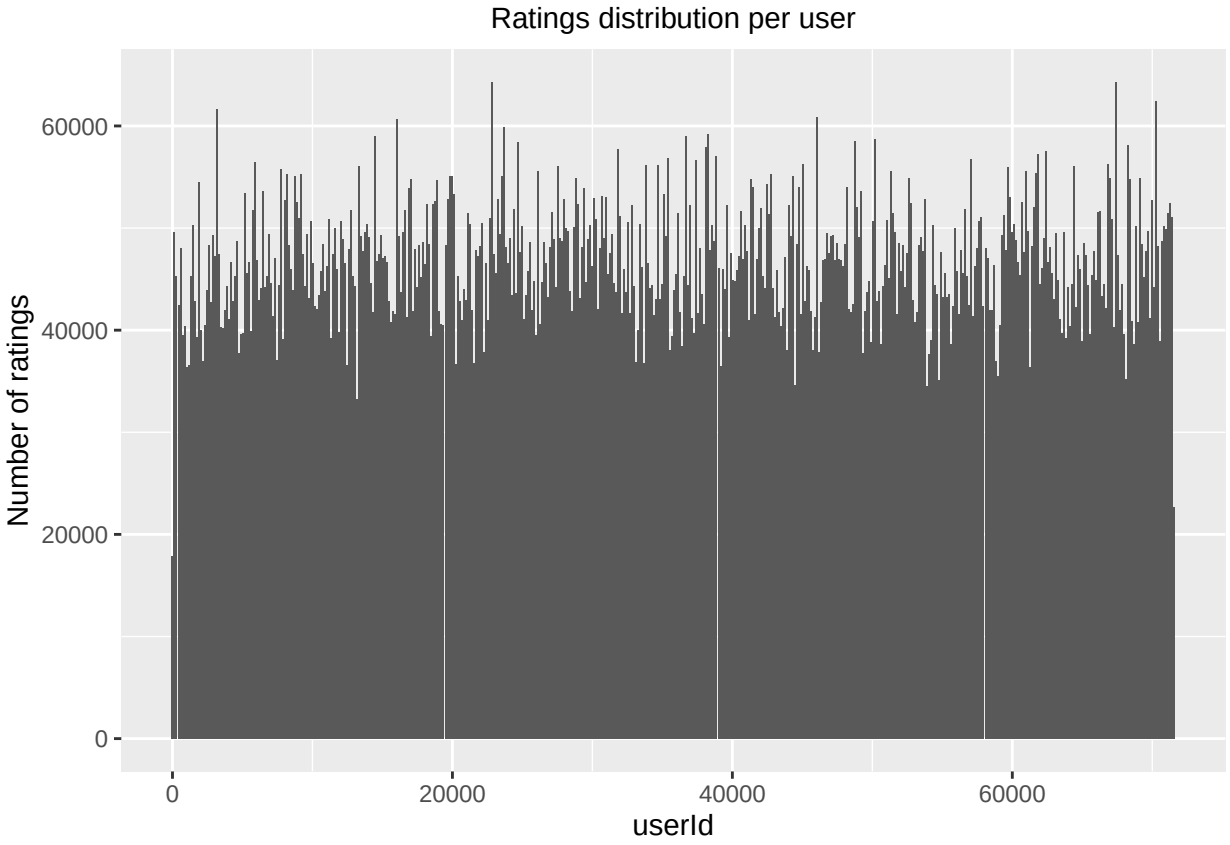


The number of ratings clearly varies from one movie to another.



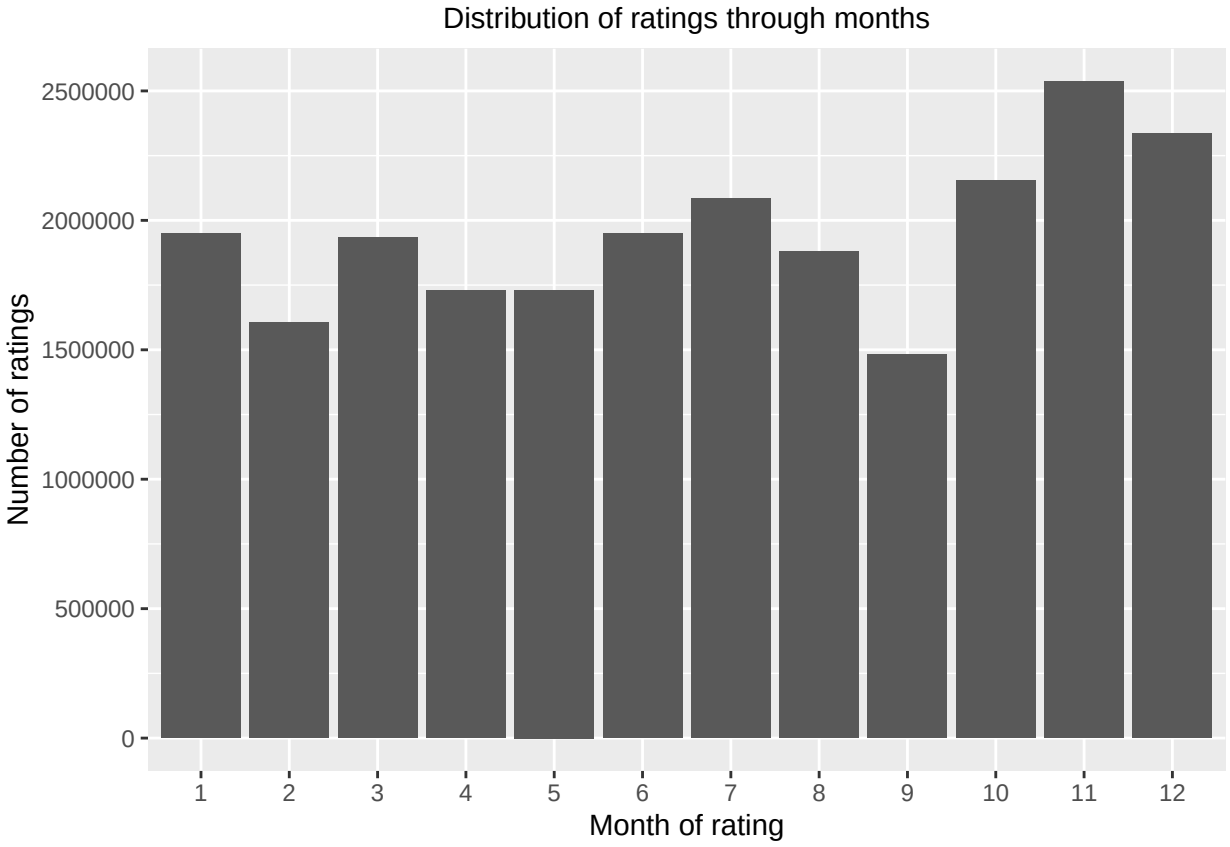
Top 50 movies cumulate up to 15.40% of the ratings in the training set.

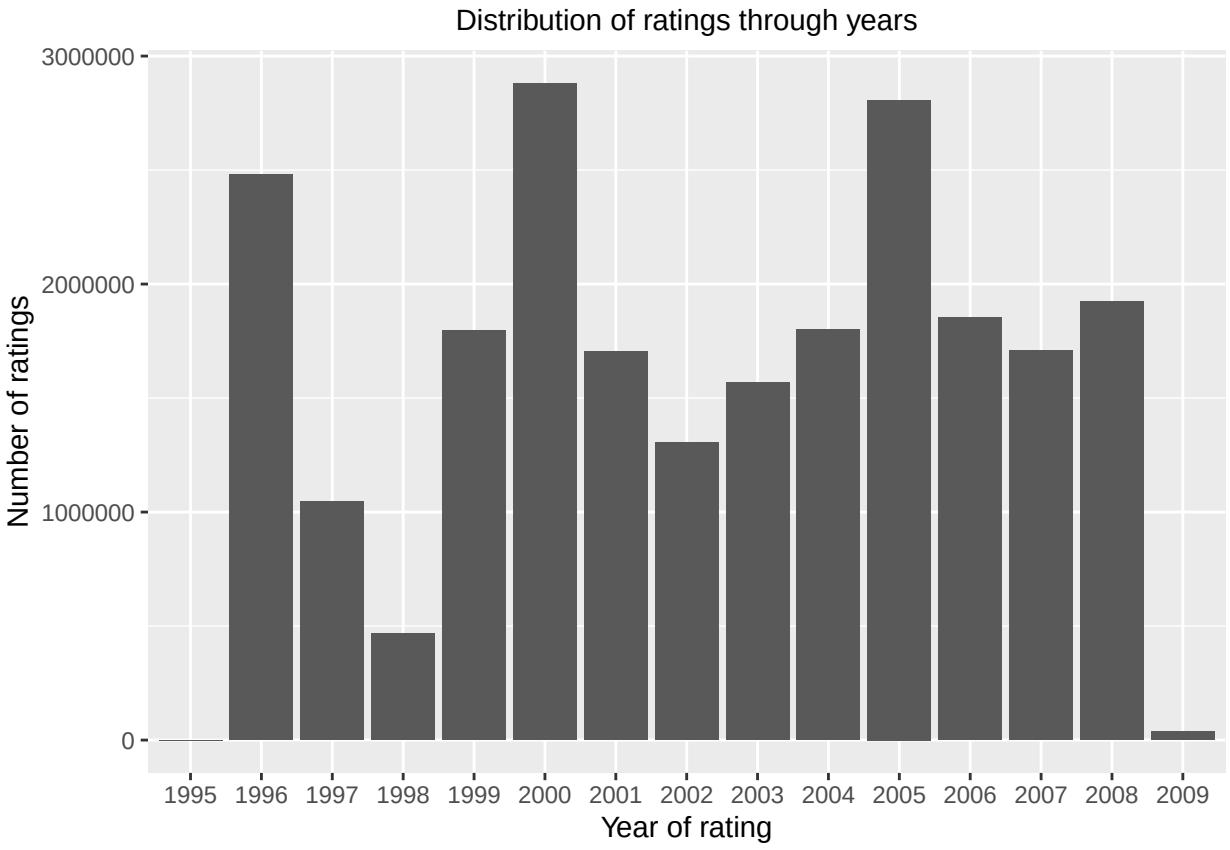
2.3.4 Ratings distribution per User



The number of ratings varies too from one user to another.

2.3.5 Ratings distribution per Month/Year of rating





2.4 Model building & evaluation

Several models will be assessed starting with the simplest. Accuracy will be evaluated using the residual mean squared error, **RMSE**. The RMSE is the error function that will measure accuracy and quantify the typical error we make when predicting the movie rating.

```
RMSE <- function(true_ratings, predicted_ratings) {
  sqrt(mean((true_ratings - predicted_ratings)^2, na.rm = TRUE))
}
```

For this case, **RMSE** value larger than **1** means that our typical error is larger than **one star**. The goal is to reduce this error below **0.8649**. Accuracies will be compared across different models.

2.4.1 Naive Mean-Baseline Model

We first started with the simplest approach we could take in making a prediction by only using the average of all movie ratings. This model is a baseline that predicts the same rating regardless of the independent variables and would look like this:

$$Y_{u,i} = \hat{\mu} + \varepsilon_{u,i}$$

Where:

- **u** is the index for users, and **i** for movies,
- $\hat{\mu}$ is the mean,
- $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0.

The **mean** is approximately **3.527** and the **RMSE** on the validation set is **1.0526**, which is very far from the target RMSE and indicates poor performance for the model.

2.4.2 Mean + Movie Effect Model

Our previous first model can be improved on by taking into account **movie bias**. Some movies are more popular than others and receive higher ratings. We can add the term b_i to reflect this **movie effect**.

$$Y_{u,i} = \hat{\mu} + b_i + \epsilon_{u,i}$$

Where:

- $\hat{\mu}$ is the mean,
- b_i is the bias of movie i, the average of $Y_{u,i}$ - $\hat{\mu}$ for each movie i.
- $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0.

The RMSE on the **validation** dataset is improved to **0.9411**, that is better than the **Naive Mean-Baseline Model**, but it is also far from the desired RMSE.

2.4.3 Mean + Movie + User Effects Model

Bias can be found in users as well. So, the **Mean + Movie + User Effects Model** consider that the users have different tastes and rate movies differently.

$$Y_{u,i} = \hat{\mu} + b_i + b_u + \epsilon_{u,i}$$

Where:

- $\hat{\mu}$ is the mean,
- b_i is the bias of movie i,
- b_u is the bias of user u,
- $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0.

The RMSE on the **validation** dataset is improved to **0.8633** that reaches the minimum performance desired. We can still hope for better results by applying the regularization techniques.

2.4.4 Mean + Movie + User + Genre Effetcs Model

Bias can also be found in movie genre as well. We can add this effect to the model as $b_{u,g}$.

$$Y_{u,i} = \hat{\mu} + b_i + b_u + b_{u,g} + \epsilon_{u,i}$$

Where:

- $\hat{\mu}$ is the mean,

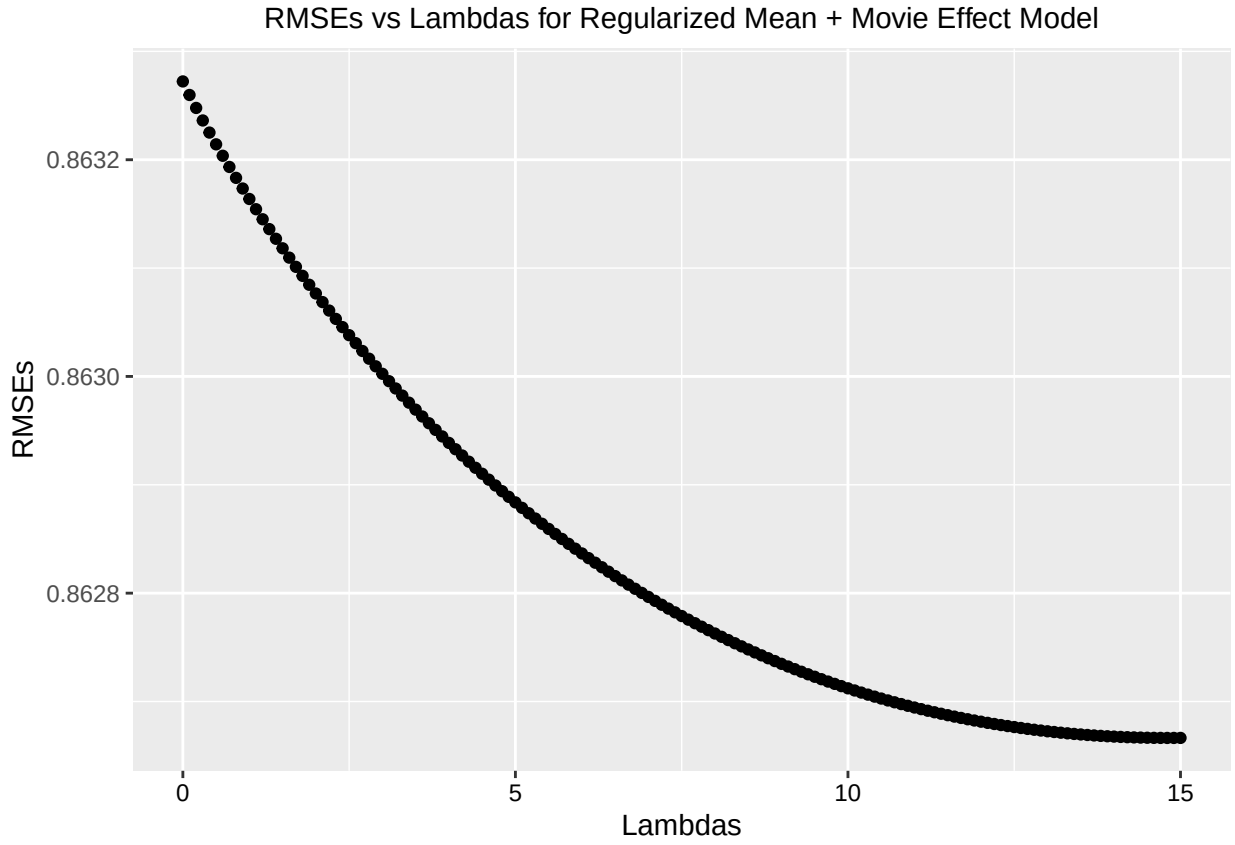
- b_i is the bias of movie i,
- b_u is the bias of user u,
- $b_{u,g}$ is the bias of movie genre, a measure for how much a user u likes the genre g,
- $\varepsilon_{i,u}$ is the independent errors sampled from the same distribution centered at 0.

The RMSE on the **validation** dataset is improved to **0.8633** that also reaches the minimum performance desired and still better. We can still hope for better results by applying the regularization techniques.

2.5 Regularization

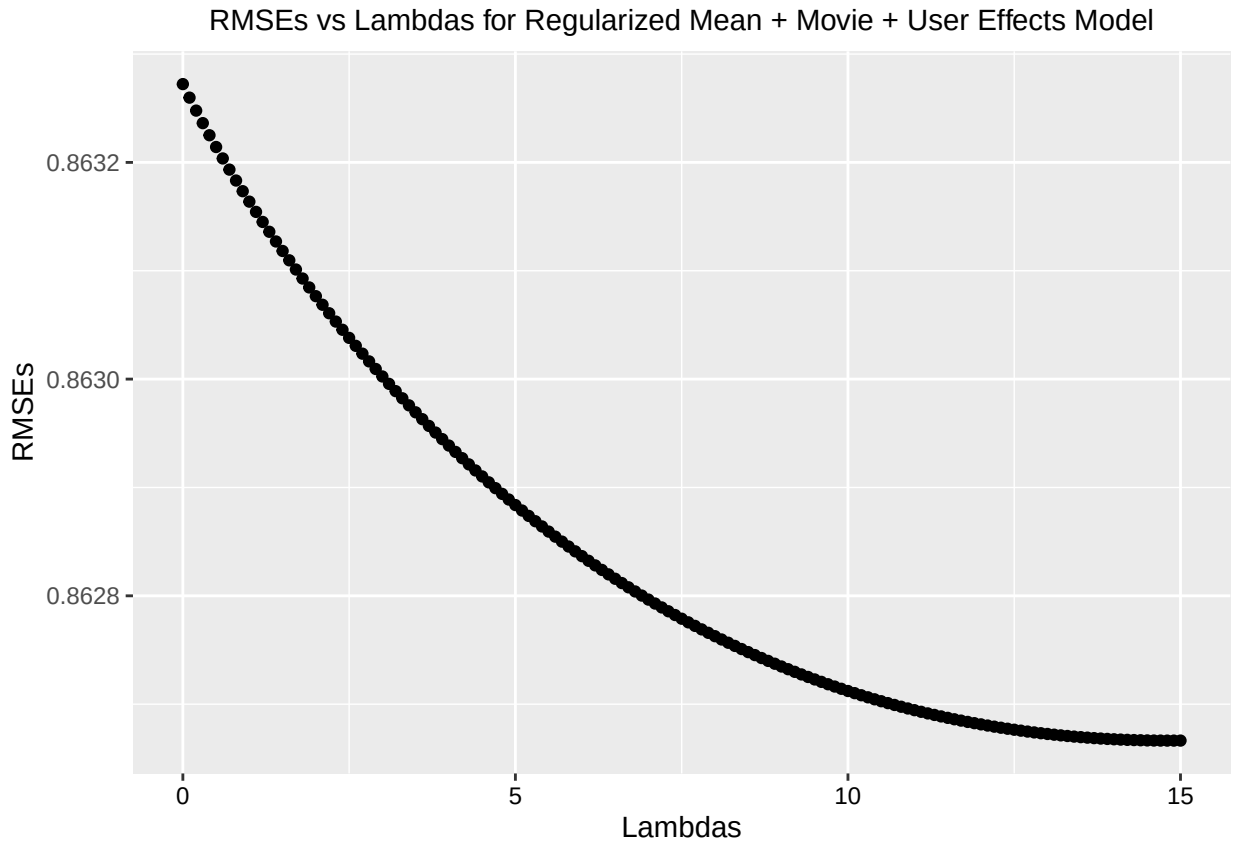
The regularization method allows us to add a penalty λ (lambda) to penalize movies with large estimates from a small sample size.

2.5.1 Regularized Mean + Movie Effect Model



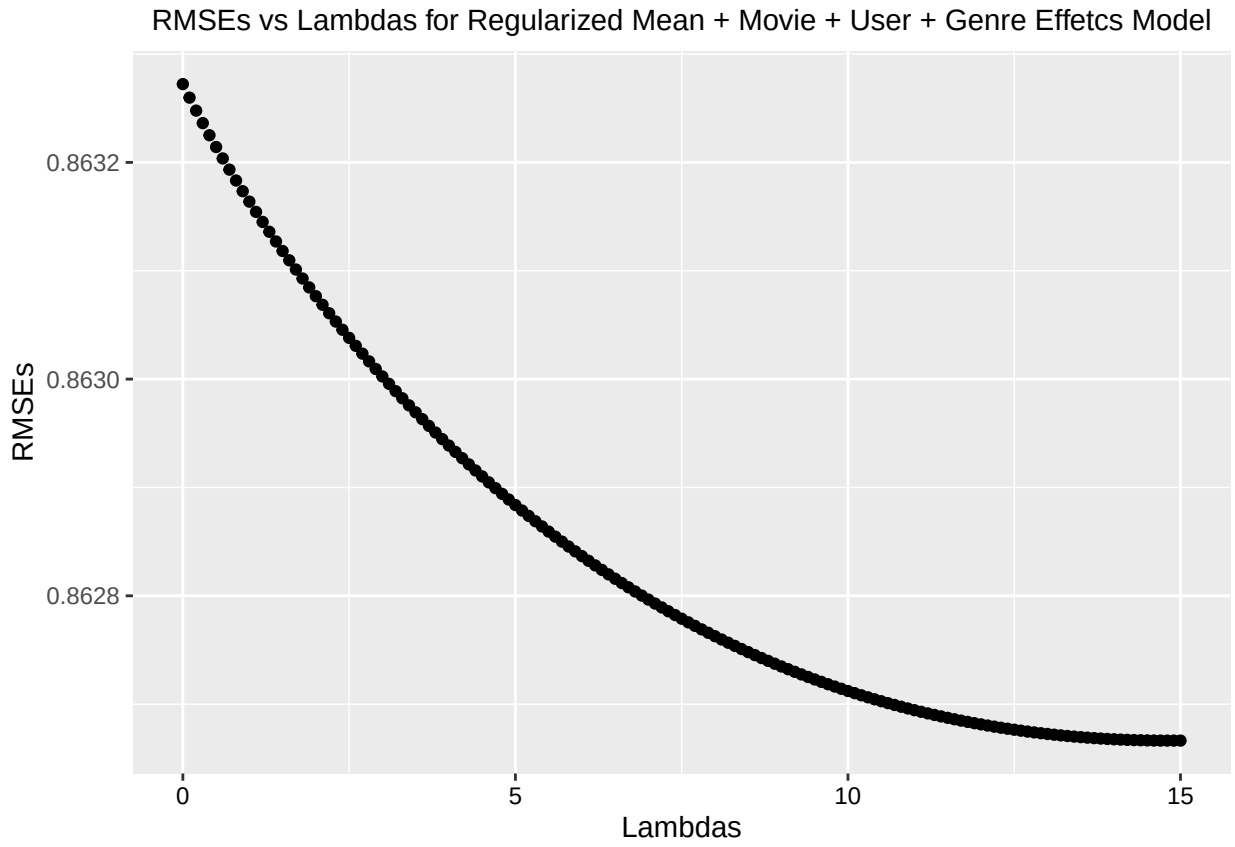
The **RMSE** on the **validation** dataset is **0.941** which is a little better than that of the **Mean + Movie Effect Model**.

2.5.2 Regularized Mean + Movie + User Effects Model



The **RMSE** on the **validation** dataset is **0.8628** which is a little better than that of the **Mean + Movie + User Effects Model**.

2.5.3 Regularized Mean + Movie + User + Genre Effetcs Model



The **RMSE** on the **validation** dataset is **0.8627** which is a little better than that of the **Mean + Movie + User + Genre Effetcs Model** and also the best result over all built models.

3 Results

To create our movie recommendation system, we buid different models considering the effects of movies, users and genres. Movies have more effects by decreasing the RMSE the most, suggesting that the movie in itself is of greatest importance to explain the rating. **Regularized Mean + Movie + User + Genre Effetcs Model** is the best off all models, achieving an RMSE of **0.8627**.

Model	RMSE
Naive Mean-Baseline Model	1.0525579
Mean + Movie Effect Model	0.9410700
Mean + Movie + User Effects Model	0.8633660
Mean + Movie + User + Genre Effetcs Model	0.8632723
Regularized Mean + Movie Effect Model	0.9410381
Regularized Mean + Movie + User Effects Model	0.8627554
Regularized Mean + Movie + User + Genre Effetcs Model	0.8626664

4 Conclusion

We buid different models considering the effects of movies, users and genres, an applying regularization. Finally, it was possible to reach a **RMSE** of **0.8627** which responds well enough to the expectations of our project.

It would have been interesting to have more predictors, informations about the users and the movies that could have made it possible to build better models.